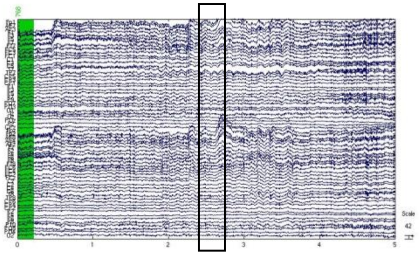
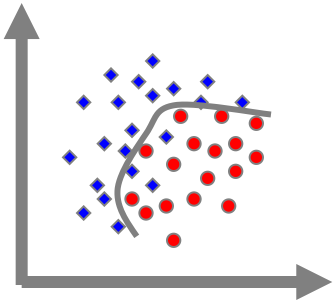
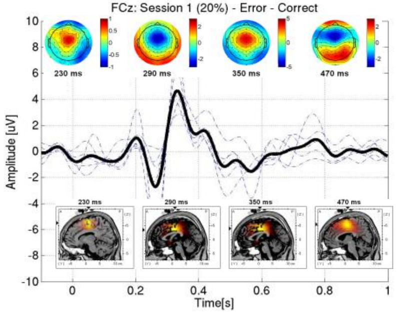






Data analysis and model classification

Week 1: MATLAB Tutorial

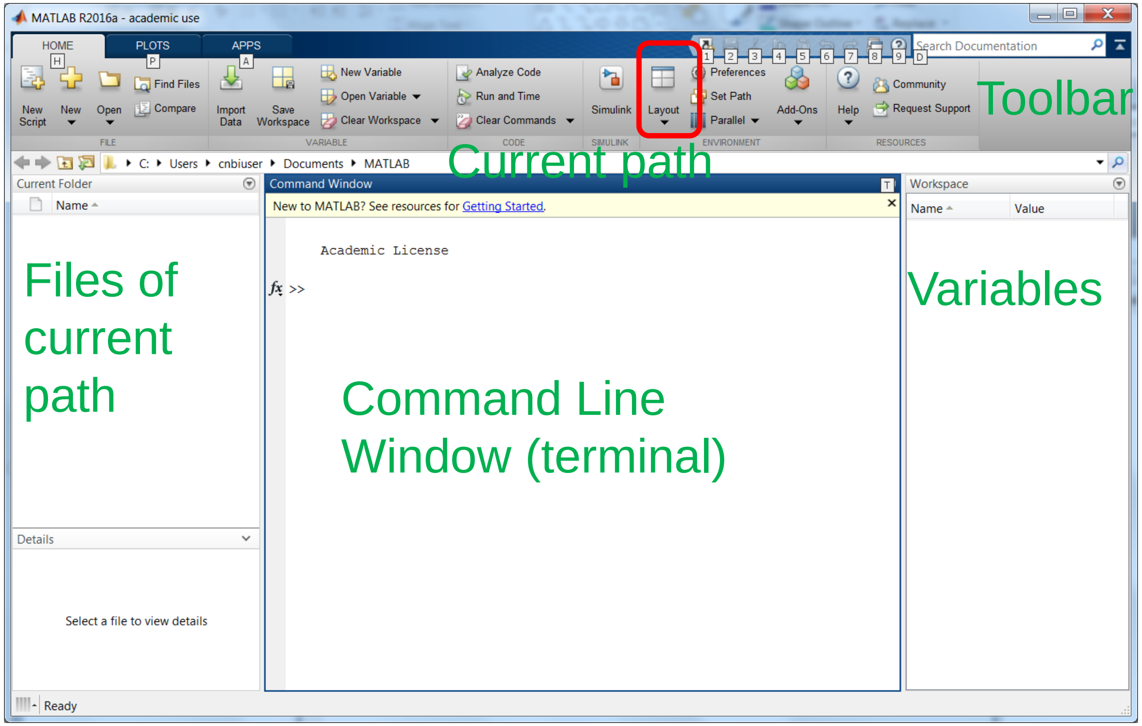
ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Getting started

- Download and install the latest available (R2016a) MATLAB from: http://poseidon.epfl.ch/pro-softwares
- Download the Tutorial from moodle
- Open MATLAB

Press for more windows and their settings, e.g.: command history, docked and so on

What you will see…



MATLAB R2016a - academic use

New to MATLAB? See resources for Getting Started.

```
Academic License

fx >>
```

Select a file to view details

Ready

Toolbar

Current path

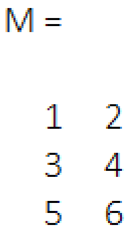
Files of current path

Command Line Window (terminal)

Variables

Basics

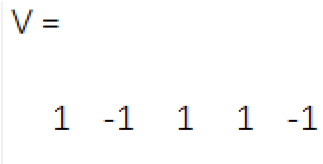
- Commonly used MATLAB commands:
  - To create a 2D matrix: >> M = [1 2; 3 4; 5 6]



```
M =

   1   2
   3   4
   5   6
```

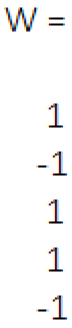
  - Similarly, for a row vector: >> V = [1 -1 1 1 -1]



```
V =

   1  -1   1   1  -1
```

  - A column vector: >> W = [1; -1; 1; 1; -1]

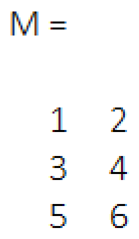


```
W =

   1
  -1
   1
   1
  -1
```

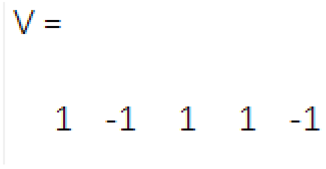
Basics

- Commonly used MATLAB commands:
  - ans is the default output variable if you don’t assign
  - Matrix dimension: >> size(M), ans = 3   2
  - Vector size: >> length(V), ans = 5
  - Maximum value: >> max(V), ans = 1
  - Minimum value: >> min(V), ans = -1
  - Assign to a variable: >> A = V or A = *function_name*(V)



```
M =

   1   2
   3   4
   5   6
```

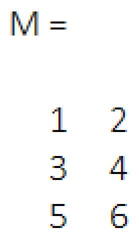


```
V =

   1  -1   1   1  -1
```

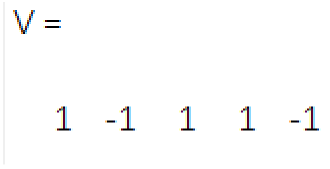
Basics

- Commonly used MATLAB commands:
  - Unique values: >> unique(V), ans = -1   1
  - Find position(s) of a value: >> find(V==1), ans = 1   3   4
  - Find number of instances of a specific value in a vector: >> length(find(V==1)), ans = 3
  - Get matrix row: >> M(1, :), ans = 1   2
  - Get matrix column: >> M(:, 1), ans = 1   3   5
  - Get partial matrix: >> M(2:3, 1:2), ans = 3   4;   5   6
  - Use of struct is unlikely to happen in this course: see example1.m for an instance on how to access and create



```
M =

     1     2
     3     4
     5     6
```

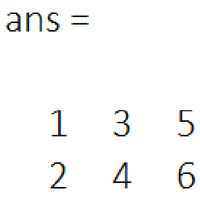


```
V =

     1    -1     1     1    -1
```

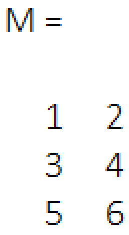
Basics

- Transpose (row -> column, column -> row)
  - >> M’
  - >> transpose(M)



```
ans =

     1     3     5
     2     4     6
```



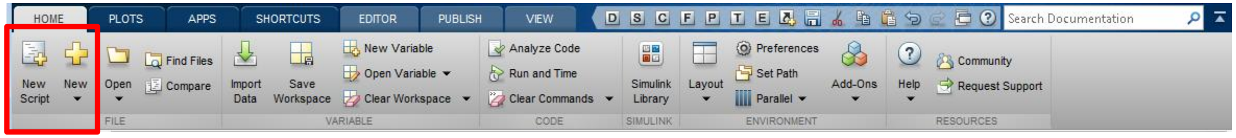
```
M =

     1     2
     3     4
     5     6
```

- Use of “;” at the end of statement for not showing the result in the command line window
  - >> a = 1

    a =

    1
  - a = 1;



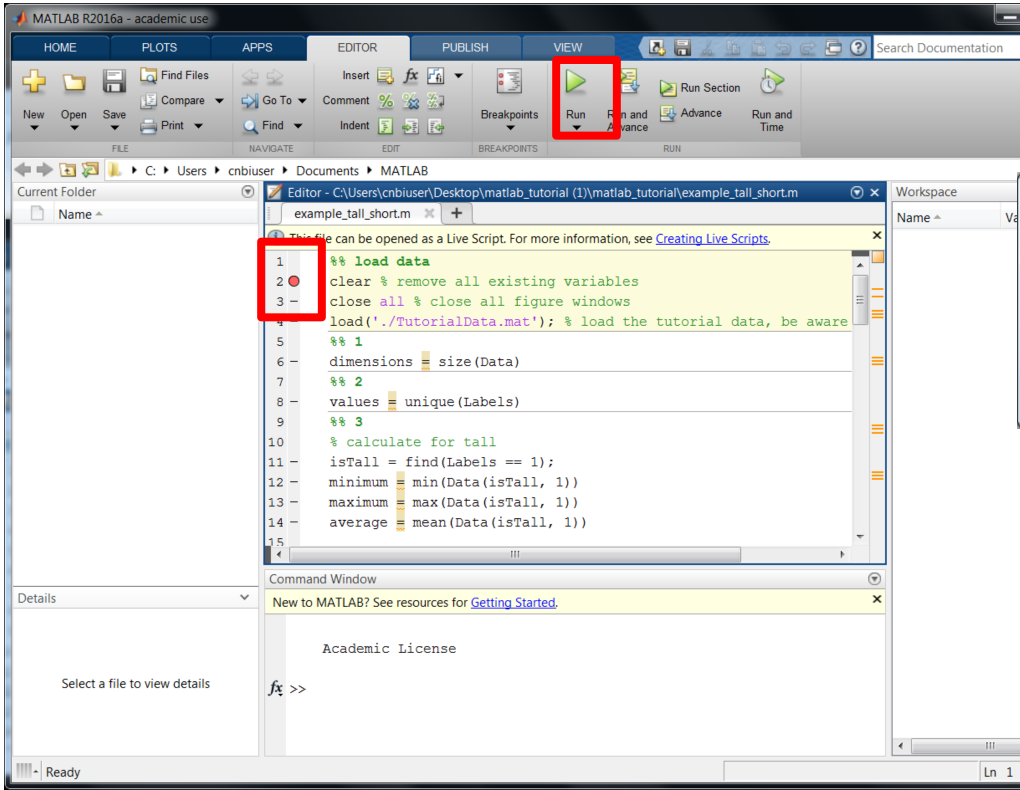
- Make a script to save your commands!

Our example

Tall vs short!

In MATLAB

- Open the script (example_tall_short.m) by double clicking the file.
- Set a breakpoint before running by clicking next to the line number
- Execute the script by clicking Run or press F5



Editor - C:\Users\cnbiuser\Desktop\matlab_tutorial (1)\matlab_tutorial\example_tall_short.m

example_tall_short.m

This file can be opened as a Live Script. For more information, see Creating Live Scripts.

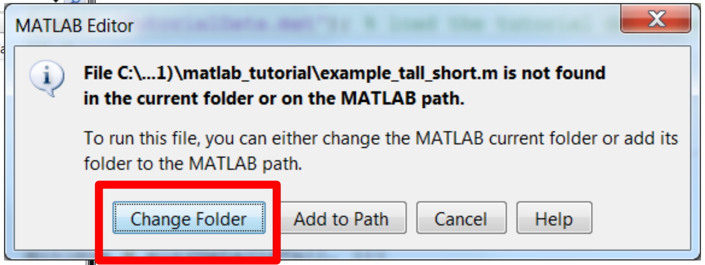
```
%% load data
clear % remove all existing variables
close all % close all figure windows
load('./TutorialData.mat'); % load the tutorial data, be aware
%% 1
dimensions = size(Data)
%% 2
values = unique(Labels)
%% 3
% calculate for tall
isTall = find(Labels == 1);
minimum = min(Data(isTall, 1))
maximum = max(Data(isTall, 1))
average = mean(Data(isTall, 1))
```

Command Window

New to MATLAB? See resources for Getting Started.

```
  Academic License

fx >>
```



MATLAB Editor

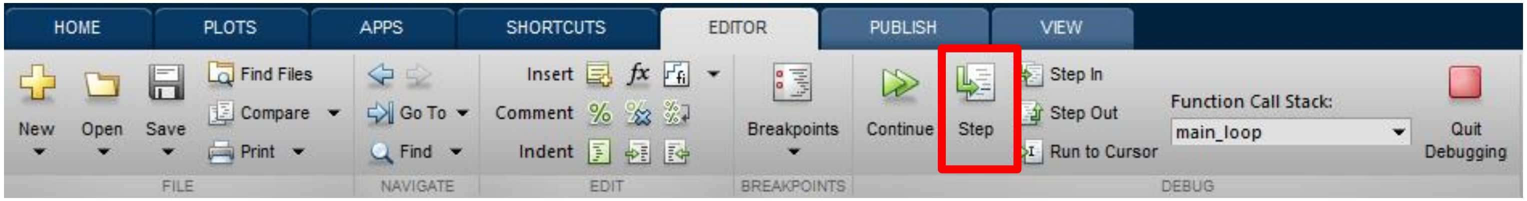
File C:\...1)\matlab_tutorial\example_tall_short.m is not found in the current folder or on the MATLAB path.

To run this file, you can either change the MATLAB current folder or add its folder to the MATLAB path.

Change folder if current path is different from the path of script file

In MATLAB

- Execute a line at a time by
  - Pressing F10
  - Click Step



- By doing so, you can learn each command in the script in a detailed way. Running without a breakpoint or step-by-step will simply give you *magic* result, you don’t know what happened

In MATLAB

- Load the variable containing the data

  Run to finish line 4 if you are using the script.

  or

  Type: ‘load TutorialData’ inside the tutorial folder (be sure that TutorialData.mat is under your current path

  or

  Drag the file into the command window

- The workspace now contains two new variables:
  - Data, dim: 10 x 3 (#samples x #features)
  - Labels, dim: 10 x 1 (#samples)

The dataset

- One person is represented by
  - ○ Height (column 1)
  - ○ Weight (column 2)
  - ○ Age (column 3)

  Height, Weight and Age are the ‘*features*’ of our problem

- We recorded the entries for 10 people

  Every person represents a ‘*sample*’ in our problem

- For every person, we say if he/she’s *tall (+1)* or *short (-1)*

  We call this description the ‘labels’ of our problem

The Exercise

1. Identify the number of samples and the number of features from the variable ‘Data’.

2. Inspect the variable ‘Labels’, identify the two integers representing if a person is tall or short.

3. Report minimum, maximum and average values of Height for all the tall people and all the short people

4. Report minimum, maximum and average values of Age for all the tall people and all the short people

size

1. Identify the number of samples and the number of features from the variable ‘Data’.

```
>> dimensions = size(Data)
dimensions =
        10    3
```

As previously said, we have 3 features (Height, Weight, Age) and 10 samples (People).

unique

2. Using the variable ‘Labels’, identify the two integers representing if a person is tall or short.

```
>> values = unique(Labels)
values =
     -1
      1
```

People can be tall (1) or short (-1).

find, min, max and mean

3. Report minimum, maximum and average values of Height for all the tall people and all the short people

Tall People

```
>> isTall = find(Labels==1);
>> minimum = min(Data(isTall,1))
minimum =
   174
>> maximum = max(Data(isTall,1))
maximum =
   185
>> average = mean(Data(isTall,1))
mean =
   178.6000
```

find, min, max and mean

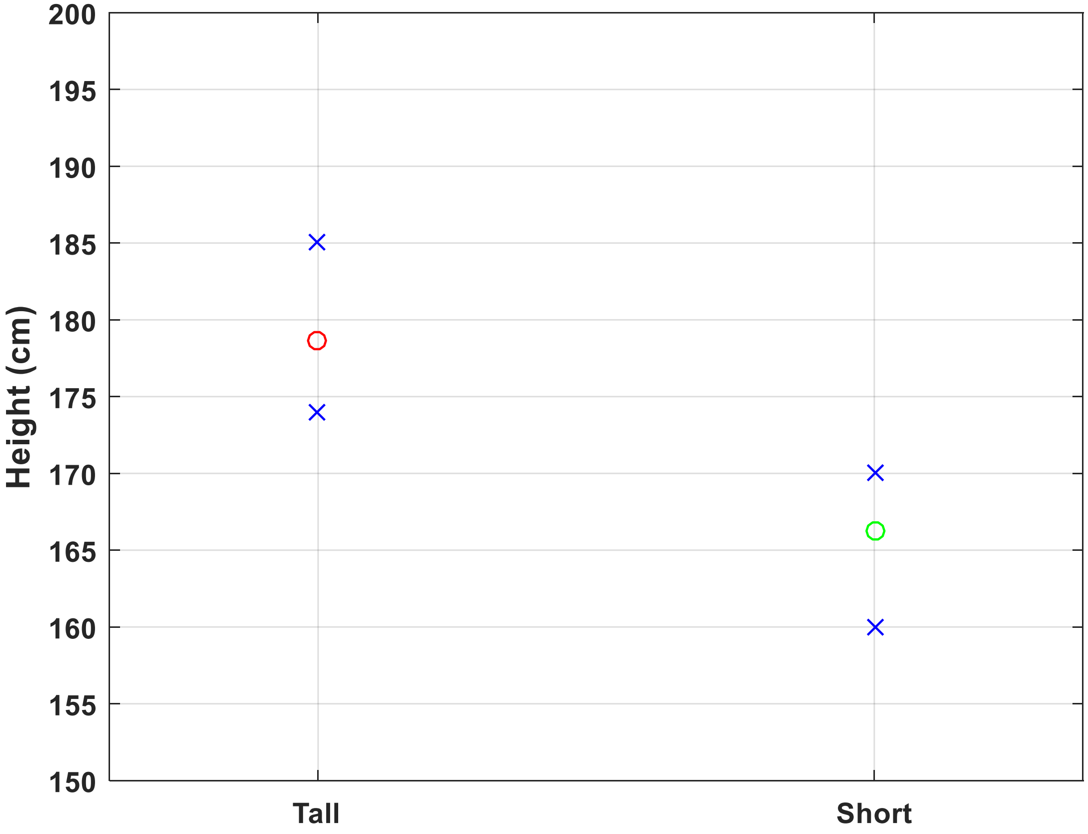
3. Report minimum, maximum and average values of Height for all the tall people and all the short people

Short People

```
>> isShort = find(Labels==-1);
>> minimum = min(Data(isShort,1))
minimum =
   160
>> maximum = max(Data(isShort,1))
maximum =
   170
>> average = mean(Data(isShort,1))
mean =
   166.2000
```

The Exercise

3. Report minimum, maximum and average values of Height for all the tall people and all the short people (read example_tall_short.m and use help, to be introduced, to learn how to plot by yourself)



find, min, max and mean

4. Report minimum, maximum and average values of Age for all the tall people and all the short people

Tall People

```
>> isTall = find(Labels==1);
>> minimum = min(Data(isTall,3))
minimum =
   18
>> maximum = max(Data(isTall,3))
maximum =
   38
>> average = mean(Data(isTall,3))
mean =
   25.6000
```

find, min, max and mean

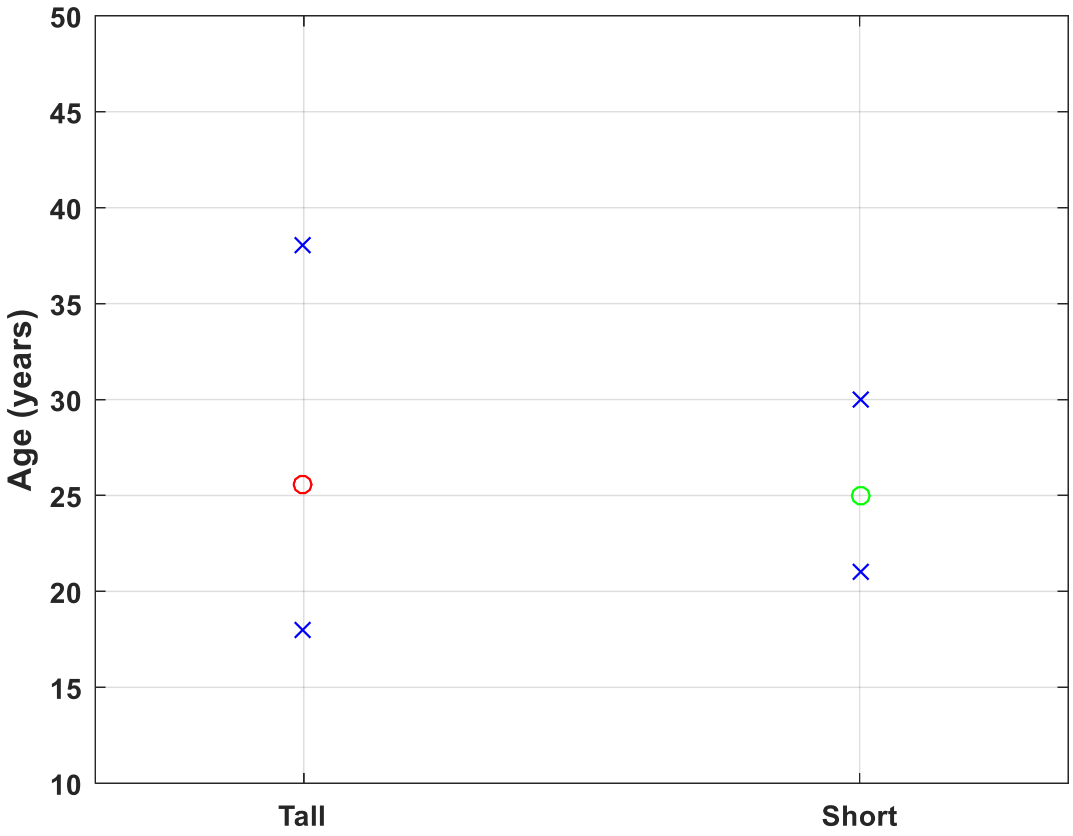
4. Report minimum, maximum and average values of Age for all the tall people and all the short people

Short People

```
>> isShort = find(Labels==-1);
>> minimum = min(Data(isShort,3))
minimum =
   21
>> maximum = max(Data(isShort,3))
maximum =
   30
>> average = mean(Data(isShort,3))
mean =
   25
```

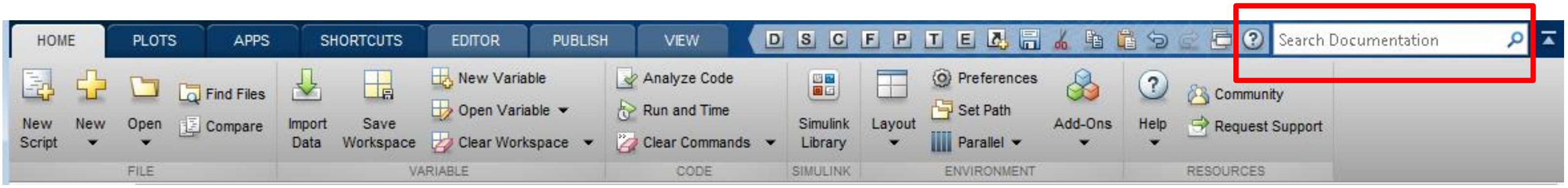
The Exercise

4. Report minimum, maximum and average values of Age for all the tall people and all the short people (read example_tall_short.m and use help, to be introduced, to learn how to plot by yourself)

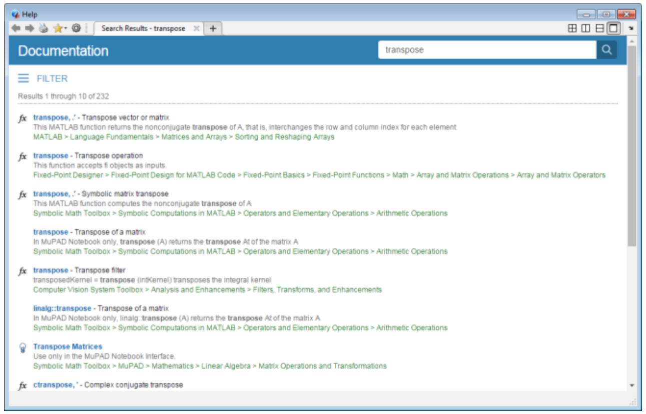


MATLAB help – from scratch

- Search from your toolbar or press F1
  - E.g.: transpose



- Google is a big hand…
- Useful websites
  - http://ch.mathworks.com/help/matlab/
  - https://ch.mathworks.com/matlabcentral/answers/
  - http://stackoverflow.com/



Results 1 through 10 of 232

fx **transpose, .'** - Transpose vector or matrix
This MATLAB function returns the nonconjugate **transpose** of A, that is, interchanges the row and column index for each element
MATLAB > Language Fundamentals > Matrices and Arrays > Sorting and Reshaping Arrays

fx **transpose** - Transpose operation
This function accepts fi objects as inputs.
Fixed-Point Designer > Fixed-Point Design for MATLAB Code > Fixed-Point Basics > Fixed-Point Functions > Math > Array and Matrix Operations > Array and Matrix Operators

fx **transpose, .'** - Symbolic matrix transpose
This MATLAB function computes the nonconjugate **transpose** of A
Symbolic Math Toolbox > Symbolic Computations in MATLAB > Operators and Elementary Operations > Arithmetic Operations

transpose - Transpose of a matrix
In MuPAD Notebook only, **transpose** (A) returns the **transpose** At of the matrix A
Symbolic Math Toolbox > Symbolic Computations in MATLAB > Operators and Elementary Operations > Arithmetic Operations

fx **transpose** - Transpose filter
transposedKernel = **transpose** (intKernel) transposes the integral kernel
Computer Vision System Toolbox > Analysis and Enhancements > Filters, Transforms, and Enhancements

linalg::transpose - Transpose of a matrix
In MuPAD Notebook only, linalg::**transpose** (A) returns the **transpose** At of the matrix A
Symbolic Math Toolbox > Symbolic Computations in MATLAB > Operators and Elementary Operations > Arithmetic Operations

Transpose Matrices
Use only in the MuPAD Notebook Interface.
Symbolic Math Toolbox > MuPAD > Mathematics > Linear Algebra > Matrix Operations and Transformations

fx **ctranspose, '** - Complex conjugate transpose

MATLAB help – given function name

- Type “*help function_name*” in command line

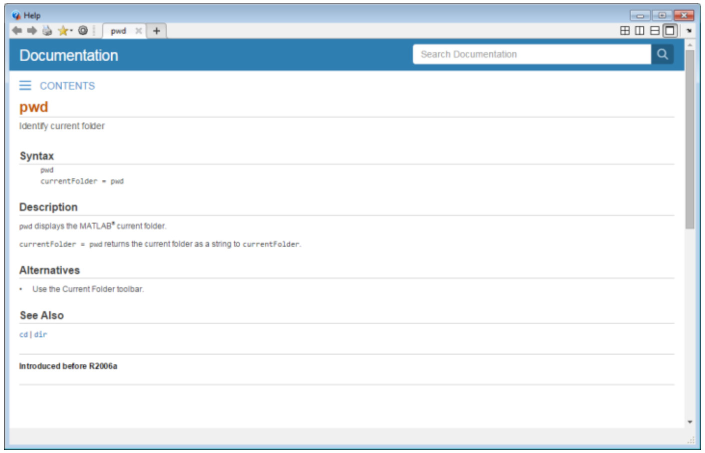
```
>> help pwd
 pwd Show (print) current working directory.
    pwd  displays the current working directory.

    S = pwd returns the current directory in the string S.

    See also cd.

    Reference page in Help browser
       doc pwd
```

- Type *“doc function_name”* to open help browser
- Type *“which function_name”* to know which function you are calling (some names are commonly used in different toolboxes)



pwd

Identify current folder

Syntax

```
pwd
currentFolder = pwd
```

Description

pwd displays the MATLAB® current folder.

currentFolder = pwd returns the current folder as a string to currentFolder.

Alternatives

- Use the Current Folder toolbar.

See Also

cd | dir

Introduced before R2006a

No Homework!

- **Try to practice if you are not familiar with MATLAB.**

Enjoy!